Touch Pattern Recognition pour l'authentification

13.05.2023

—

Diarra Yacouba, Kanté Younouss Koly, Haïdara Mohamed,

Diabaté Maïmouna, Maïga Maïmouna Henné

@Institut Privé Africain de Technologie et de Management (ITMA) : Non impliqué dans la recherche

Baco Djicoroni ACI, Bamako, Mali

Baco Djicoroni ACI, Bamako, Mali

Aperçu

Ce rapport intermédiaire vise à être un bon point de contrôle sur la progression de ce projet.
Le projet Touch Pattern Recognition est un projet qui aspire à révolutionner les systèmes d'authentification. Ce projet émane de cinq étudiants de l'université @ITMA Diarra Yacouba, Kanté Younouss Koly, Haïdara Mohamed, Diabaté Maïmouna et Maïga Maïmouna Henna dans le cadre de la deuxième édition de « TRANSFORMATION NUMERIQUE », le concours organisé par Africa_Digital en mai 2023.
Ces étudiants ont participé à la ligue des universités "TRANSFORMATION NUMERIQUE" qui portait sur la sécurité des données.
Notez que ce fichier peut être modifié à tout moment en raison de l'état d'avancement du projet et que tous les événements mentionnés dans ce document sont localisés à Bamako/Mali entre le 30 avril et le 13 mai 2023.

Cela dit, l'idée derrière le projet est de remplacer les identifiants d'authentification système traditionnels (nom d'utilisateur/mot de passe) par un modèle d'apprentissage automatique (ML) formé pour reconnaître l'utilisateur du système via des données tactiles sur des appareils tactiles.
À cette fin, les données sont représentées avec différents champs tels que les positions, les vitesses, les directions, la latence (délai) de chaque touche.

Ce document passera en revue le projet, de l'acquisition des données au déploiement du modèle en passant par les problèmes et problèmes rencontrés.

Buts

Ce projet vise à :

1. Repenser notre façon de penser les systèmes d'authentification.
2. Apporter plus d'innovation et de sécurité aux systèmes d'authentification.
3. Facilitez l'accès aux systèmes en remplaçant les mots de passe qui peuvent être fastidieux car ils sont faciles à oublier ou à pirater.
4. Rechercher de nouvelles applications des systèmes d'IA qui deviennent de plus en plus puissant.

Acquisition et prétraitement des données

Nous allons parler de la manière dont les données ont été collectées et traitées pour la formation.

I. Acquisition de données

Tout d'abord, nous avons utilisé une application Web mobile existante appelée "TouchTracker" développée par Marco Cancellieri. Cette application possède une interface utilisateur simple qui affiche un écran vert et enregistre tous les gestes effectués sur l'écran écrit en JavaScript.
Plus d'infos sur : TouchTracker

Cependant, l'objectif initial de l'application n'était pas de collecter les données tactiles dont nous avons besoin, nous avons donc apporté des modifications à son code source pour collecter les données nécessaires.
Plus précisément, vous avez modifié le code de l'application pour collecter les positions tactiles, les vitesses, les directions, les durées et les horodatages.

Ces modifications vous ont permis de recueillir des informations détaillées sur les gestes tactiles effectués sur l'écran, ce qui est essentiel pour entraîner un modèle d'apprentissage automatique à reconnaître et à faire des prédictions avec précision.

Une fois les données collectées, elles ont été exportées au format de fichier .json. Il s'agit d'un format de données structuré qui est facilement lisible et traité par la plupart des langages de programmation, y compris Python, le langage que nous avons utilisé pour entraîner votre modèle d'apprentissage automatique.

Pour commencer le processus d'acquisition de données, nous avons créé une interface utilisateur pour notre site Web basé sur TouchTracker. Nous l'avons fait en modifiant un autre code source initialement écrit pour les événements de souris. Afin d'adapter ce code pour qu'il fonctionne avec les événements tactiles, nous avons dû modifier le code HTML, CSS et JS.

HTML, CSS et JS sont les trois technologies fondamentales utilisées pour créer des sites Web et des applications Web. HTML fournit la structure de la page, CSS contrôle la présentation et la mise en page, et JS gère l'interactivité et la fonctionnalité.

Nous devions modifier le code JS pour gérer les événements tactiles au lieu des événements de souris. Cela impliquait probablement de réécrire ou de modifier les écouteurs d'événements pour reconnaître les gestes tactiles tels que les tapotements, les balayages et les pincements, et pour collecter les données tactiles dont nous avions besoin pour le modèle d'apprentissage automatique.

Dans l'ensemble, votre processus d'acquisition de données impliquait la modification d'applications Web mobiles existantes, la collecte de données tactiles détaillées et l'exportation des données dans un format compatible.
Ces étapes fournissent une base solide pour la formation d'un modèle d'apprentissage automatique afin de reconnaître et de prédire les propriétaires tactiles.

Notez qu'à la date de rédaction de ce rapport, l'application n'était pas entièrement terminée et n'était qu'un prototype après jusqu'à 8 jours d'édition de code en raison de problèmes qui seront décrits dans la dernière section de ce document.

II. Prétraitement Après

l'acquisition des données, nous devions représenter les données sous une forme appropriée pour former un modèle d'apprentissage automatique. À cette fin, nous avons écrit un script python qui ouvre et lit chaque fichier .json téléchargé et en extrait les champs qui nous intéressent, tels que les positions (liste de 2 listes d'éléments où la première est Position.X et la seconde Position.Y) , les vitesses (liste des vitesses euclidiennes calculées pour chaque nouvelle position enregistrée et divisées à chaque fois par le temps écoulé depuis le déclenchement de l'écouteur d'événement touchstart), les directions (liste des directions/angles de chaque touche par rapport à la précédente en radians) et les horodatages que nous avons utilisés pour calculer la durée de chaque contact.

Notez que les positions ont été divisées en deux nouvelles listes position.X et position.Y.

Et puis afin d'obtenir les fonctionnalités qui serviront à former le modèle nous

calculé la moyenne, la dérivation standard, min, max et la plage de chaque champ ou listes (Position.X, Position.Y, vitesses, directions) et ajouté la durée du toucher.

De cette façon, nous avons obtenu une nouvelle représentation de nos données qui est un numpy.array de forme (21,) qui est une très bonne représentation pour entraîner un modèle de Deep Learning.

Les données ont ensuite été étiquetées et enregistrées avec Pyhton.pickle .

NB : L'étiquette était des valeurs binaires (0, 1)

Formation et évaluation de l'IA

I. Création & Formation

Nous avons décidé d'opter pour un modèle Keras.

Keras est une bibliothèque Python pour l'apprentissage en profondeur qui enveloppe les bibliothèques numériques efficaces Tensorflow et Théano. Keras vous permet de concevoir et d'entraîner rapidement et simplement des réseaux de neurones et des modèles d'apprentissage en profondeur.

Nous créons donc un Keras Sequential assez simple modèle.

Les poids sont initialisés à l'aide d'un petit nombre aléatoire gaussien. La fonction d'activation du redresseur est utilisée. La couche de sortie contient un seul neurone afin de faire des prédictions. Il utilise la fonction d'activation sigmoïde afin de produire une sortie de probabilité dans la plage de 0 à 1 qui peut facilement et automatiquement être convertie en valeurs de classe nettes.

Enfin, vous utiliserez la fonction de perte logarithmique (cross-entropy) pendant la formation, la fonction de perte préférée pour les problèmes de classification binaire. Le modèle utilise également l' optimiseur efficace d'Adam pour la descente en pente, et les métriques de précision seront collectées lors de la formation du modèle.

Le modèle n'avait qu'une seule couche cachée de 10 neurones. Ainsi, la topologie du modèle peut être résumée par [21, 10, 1]. Il n'avait que 693 paramètres

II. Évaluation et sauvegarde

Nous avons d'abord évalué les topologies de modèles de couple à l'aide d'une validation croisée stratifiée dans le cadre scikit-learn.

L'idée était d'utiliser cette étape comme un test consistant à s'entraîner pendant une courte période et à les évaluer afin de choisir le meilleur modèle pour s'entraîner plus longtemps.

La validation croisée stratifiée (StratifiedKFold) est une technique de rééchantillonnage qui fournira une estimation des performances du modèle. Pour ce faire, il divise les données en k parties et entraîne le modèle sur toutes les parties sauf une, qui est présentée comme un ensemble de test pour évaluer les performances du modèle.

Faute de temps, nous avons failli sauter cette étape. Nous avons été vraiment surpris par les performances du modèle même sans aucune couche cachée qui a marqué 83%
précision.

Après une formation supplémentaire de seulement 1000 époques, la topologie [21, 10, 1] a atteint une précision de 90 % et c'est le modèle que nous avons enregistré au moment de la rédaction de ce document (13 mai 2023).

Nous avons ensuite enregistré les modèles pré-formés prêts pour l'inférence, le test ou la formation complémentaire dans différents formats :

- Dans un dossier

Le format SavedModel est un répertoire contenant un binaire protobuf et un point de contrôle
TensorFlow. La première approche utilisée pour enregistrer le modèle

- Au format TF.js Layers

Nous avons enregistré le modèle au format TF.js Layers après un bref test pour l'utiliser ultérieurement avec l'application

Déploiement du modèle

C'est l'étape réelle sur laquelle nous travaillons.

Nous avons déjà exporté le modèle vers Tensorflow.js couches et nous essayons de l'intégrer à l'application Web utilisée pour collecter les données, mais nous rencontrons en fait des problèmes et des problèmes d'importation liés à TFJS et aux événements dom. Plus de détails dans la section des problèmes

Questions

Nous avons eu un délai de 2 semaines pour faire ce rapport partiel.
Nous avons rencontré quelques problèmes, la plupart d'entre eux ont été résolus ou étaient plus des erreurs de programmation que de vrais problèmes ou bugs.

Dans cette section, nous allons parler de deux problèmes / bogues qui n'ont pas encore été corrigés.

I. Mouseenter => Événements Touchenter

La finition de l'interface utilisateur de l'application Web implique de trouver des alternatives aux événements de souris dans les événements tactiles. Dans une partie du code, nous devons faire cette conversion, mais il semble que l'événement toucherer n'agisse pas vraiment comme le fait mouseenter pour les événements de souris.

L'événement mouseenter se déclenche à chaque fois que la souris entre dans la zone d'un élément du HTML tandis que l'événement toucherer se déclenche lorsque le toucher démarre sur l'élément comme l'événement touchstart qui est souvent utilisé à la place de mouseenter. Nous travaillons actuellement sur une solution pour résoudre ce problème. Si besoin nous envisageons de réécrire le code pour ne plus être dépendant de ces évènements

II. Importation de Tensorflow.js dans un module utilisant des événements dom

Nous avons remarqué que l'importation de Tensorflow.js dans un module JavaScript qui utilise des événements dom empêchera ces événements de fonctionner.

Nous travaillons donc également sur ce problème afin d'intégrer le modèle en JavaScript et de l'utiliser dans le navigateur.

A cet effet Carlos a proposé de nous aider, sachant qu'il avait contourné ce problème il y a quelques années avec une autre version de Tensorflow.js. Son aide nous sera précieuse et nous pourrons peut-être résoudre ou contourner rapidement ce problème.

Résumé et remarques

Le résumé de ce document présentait un petit projet de recherche en cours mené par cinq étudiants au Mali.

Avec les résultats décrits ici, nous pensons que ce projet a un grand potentiel et aimerions intéresser les gens.

Nous espérons que cela pourra être un point de départ pour de grands progrès dans les domaines de l'IA et de la cybersécurité et de l'informatique à l'échelle mondiale.

En espérant que nous pourrions participer au futur de la technologie avec ce projet.

Un rapport final sera fait une fois le projet terminé, il n'y a rien de spectaculaire pour l'instant mais j'espère que ce sera le cas.

Vous pouvez obtenir tous les codes écrits à cette date sur ce github et contacte les auteurs avec les liens suivants :

Kanté Younouss Koly : email

Haïdara Mohamed : email, Facebook, LinkedIn,

Diabaté Maïmouna : email

Diarra Yacouba : courriel, Facebook, LinkedIn, Twitter, github, Débordement de pile

Maïga Maïmouna Henné : email, Twitter, LinkedIn